

Tomasz PAJCHROWSKI\*

## ADAPTACYJNE STEROWANIE SILNIKIEM SYNCHRONICZNYM O MAGNESACH TRWAŁYCH

W artykule przedstawiono wyniki symulacyjne adaptacyjnego regulatora prędkości z zastosowaniem sztucznej sieci neuronowej dla napędu z silnikiem synchronicznym o magnesach trwałych. Omówiona została struktura sztucznej sieci neuronowej oraz metoda uczenia regulatora prędkości w czasie rzeczywistym. Model układu został opracowany w języku Matlab. Parametry regulatora są optymalizowane on-line według algorytmu RPROP. Przedstawione wyniki badań symulacyjnych ilustrują poprawne działanie adaptacyjnej regulacji prędkości na zmianę parametrów układu napędowego, takich jak moment bezwładności.

### 1. WPROWADZENIE

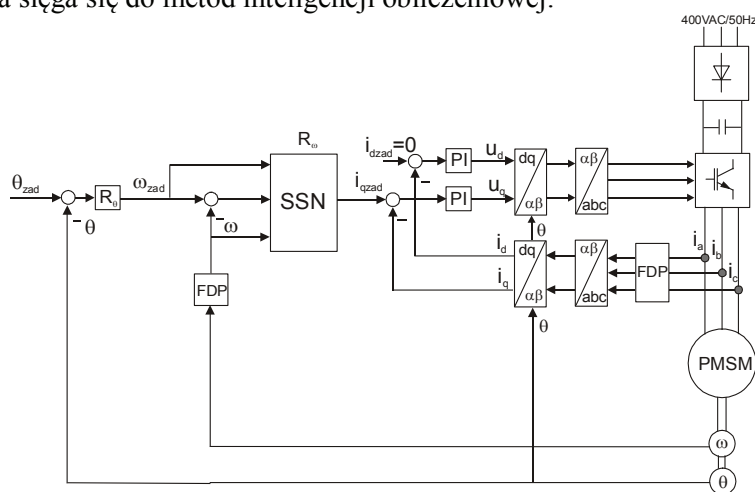
Współczesnym układom napędowym, zwłaszcza serwonapędom, stawia się coraz wyższe wymagania dynamiczne, co powoduje, że parametry regulatorów prędkości i położenia stosowane w tych układach są dobierane tak, aby uzyskać największą, możliwą dynamikę ruchu. Tak dobrane regulatory o „wyśrubowanych” nastawach są wrażliwe na niewielkie nawet zmiany parametrów transmitancji napędu i napędzanego mechanizmu. W takich układach napędowych jak: napędy ramion robota, napędy mechanizmu posuwu obrabiarek, napędy zwijarek czy maszyn papierniczych, parametrem najczęściej zmienianym jest moment bezwładności. Zmienne jest również opóźnienie występujące w układach przekształtników i sterowania, jak i również w stanach osłabiania strumienia magnetycznego zmienna jest stała momentu elektromagnetycznego silnika. Dlatego zachodzi potrzeba zaprojektowania układu regulacji niewrażliwego, lub mało wrażliwego na zmiany wyżej wymienionych parametrów.

Na rys. 1 przedstawiono strukturę kaskadowej regulacji serwonapędu z silnikiem synchronicznym o magnesach trwałych (*ang. PMSM- Permanent Magnet Synchronous Magnet*), składającą się z pętli regulacji momentu elektromagnetycznego, prędkości i położenia. Silnik ten, ze względu na mały moment bezwładności, dużą przeciążalność momentem i wysoką sprawność, jest bardzo chętnie stosowany w przemyśle i stawiany w rzędzie najlepszych pod względem dynamiki silników. Dla uzyskania małej wrażliwości na zmiany

---

\* Politechnika Poznańska.

parametrów całego układu regulacji, kluczowym staje się zaprojektowanie odpornego lub adaptacyjnego regulatora prędkości kątowej, przy założeniu, że nie zmieniają się parametry w pętli regulacji momentu, natomiast pętla regulacji położenia tych zmian może nie odczuwać przy prawidłowym zaprojektowaniu regulatora prędkości. Obecnie bardzo chętnie, przy projektowaniu takiego regulatora sięga się do metod inteligencji obliczeniowej.



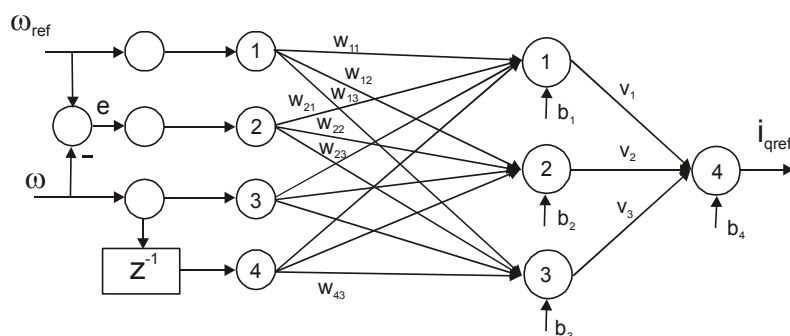
Rys. 1. Schemat układu regulacji położenia i prędkości dla serwonapędu z PMSM

W pracach wcześniejszych [8, 9] autor brał udział w opracowaniu koncepcji odpornych regulatorów prędkości wykorzystujących logikę rozmytą, sztuczne sieci neuronowe i połączeniu obu tych technik w systemach neuronowo-rozmytych. Efektem działania regulatorów odpornych było uzyskanie jednakowej, niezależnej od zmian momentu bezwładności dynamiki regulacji prędkości. Alternatywną koncepcją dla sterowania odpornego jest regulacja adaptacyjna. W pracach [2, 3, 4] przedstawiono dwie różne koncepcje adaptacyjnego regulatora neuronowego. Regulator adaptacyjny z modelem referencyjnym przedstawiono w pracy [4], w którym regulator neuronowy uczony jest na podstawie błędu, jaki powstaje pomiędzy wzorcowym sygnałem wyjściowym z modelu referencyjnego, a rzeczywistym sygnałem regulowanego obiektu. Inną ciekawą koncepcję, w której regulator neuronowy uczony jest na podstawie własnego błędu regulacji zaprezentowano w pracach [2, 3]. W pracach tych autorzy zaprezentowali oryginalne rozwiązania tej metody, które nadal wydaje się wymagać dalszej pogłębionej analizy. W niniejszym artykule autor przedstawił wyniki badań symulacyjnych układu regulacji prędkości obrotowej na zmianę momentu bezwładności. Badania przeprowadzono dla serwonapędu bezpośredniego (bez przekładni mechanicznej) z wolnoobrotowym silnikiem PMSM, którego parametry podano w [6, 7, 8, 9].

## 2. STRUKTURA I ALGORYTM UCZENIA SIECI NEURONOWEJ

### 2.1. Struktura sieci neuronowej

Na rysunku 2 przedstawiono przyjętą strukturę sztucznej sieci neuronowej, która w układzie regulacji prędkości pełni rolę regulatora adaptacyjnego.



Rys. 2. Struktura sztucznej sieci neuronowej

Sieć ta posiada trzy wejścia główne dla sygnałów prędkości zadanej ( $\omega_{ref}$ ), prędkości rzeczywistej ( $\omega$ ) i uchybu ( $e$ ), oraz wejście dodatkowe sygnału uchybu opóźnionego o jeden okres próbkowania, który w modelu wynosił  $100 \mu s$ . Sieć posiada dwie warstwy o liniowych funkcjach aktywacji, gdzie sygnałem wyjściowym jest prąd zadany w osi  $q$  ( $i_{qref}$ ).

### 2.2. Algorytm uczenia sieci neuronowej

Uczenie sztucznej sieci neuronowej odbywa się bez sygnału wzorcowego, na podstawie uchybu regulacji prędkości. Jako kryterium jakości uczenia przyjęto wyrażenie [5, 11]:

$$E = \frac{1}{2} (\omega_{ref} - \omega)^2 \quad (1)$$

Ponieważ adaptacja wag regulatora neuronowego ma się odbywać w czasie rzeczywistym (*ang. on-line*), należy wybrać prosty i szybki algorytm modyfikacji parametrów sztucznej sieci neuronowej. Po analizie literatury i stosowanych rozwiązaniach wybrana została metoda RPROP (*ang. Resilient backPROPagation*) [10], która wymaga jedynie znaku składowej gradientu, bez określania ich wartości, a współczynniki uczenia są modyfikowane w każdym kroku uczenia. Zmiana współczynnika wagi  $\Delta w_{ij}(k)$  dla  $j$ -tego wejścia  $i$ -tego neuronu opisuje zależność [10]:

$$\Delta w_{ij}(k) = -\eta_{ij}(k) \cdot \text{sign} \left( \frac{\partial E(w(k))}{\partial w_{ij}} \right) \quad (2)$$

Współczynnik uczenia  $\eta_{ij}(k)$  jest indywidualnie dobierany dla każdej wagi, i zwiększany, gdy znaki gradientów są takie same, natomiast zmniejszany, gdy są różne, według następującej zasady [10]:

$$\eta_{ij} = \begin{cases} \min(a\eta_{ij}(k-1), \eta_{\max}) & \text{dla } S_{ij}(k)S_{ij}(k-1) > 0 \\ \max(b\eta_{ij}(k-1), \eta_{\min}) & \text{dla } S_{ij}(k)S_{ij}(k-1) < 0 \\ \eta_{ij}(k-1) & \text{dla innych przypadków} \end{cases} \quad (3)$$

gdzie  $\eta_{\min}$ ,  $\eta_{\max}$  oznacza minimalną i maksymalną wartość współczynnika uczenia,  $a$  i  $b$  wartości stałe (najczęściej  $a = 1,2$ ,  $b = 0,5$ ) a

$$S_{ij}(k) = \frac{\partial E(w(k))}{\partial w_{ij}} \quad (4)$$

Zaletą tej metody uczenia jest znaczne przyspieszenie procesu modyfikacji wag, szczególnie w obszarach o niewielkim nachyleniu funkcji celu [10].

### 3. METODYKA BADAŃ SYMULACYNYCH

Badania symulacyjne przeprowadzono w środowisku Matlab – Simulink, wykorzystując opracowany model układu napędowego z silnikiem PMSM, opisany w pracach [8, 9]. Sztuczną sieć neuronową zmodelowano wykorzystując program narzędziowy Neural Networks Toolbox. W pracy założono, że w układzie napędowym brak jest połączeń sprzężystych i zmian energii potencjalnej dlatego można zapisać:

$$m_d \omega = \frac{d}{dt} \left( \frac{J\omega^2}{2} \right) \quad (5)$$

a następnie po przekształceniach uzyskujemy wyrażenie:

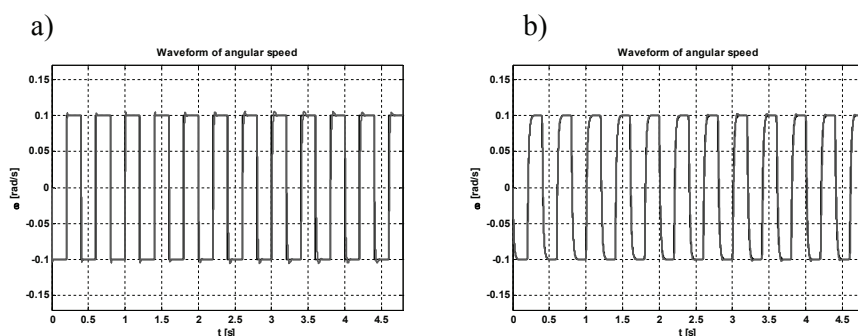
$$m_d = J \frac{d\omega}{dt} + \frac{\omega}{2} \frac{dJ}{dt} \quad (6)$$

Po zbudowaniu modelu matematycznego przeprowadzono badania, które polegały na wymuszaniu procesów przejściowych wywołanych zmianą prędkości zadanej dla różnych wartości momentu bezwładności  $J$ , którego wartości zmieniały się w funkcji czasu.

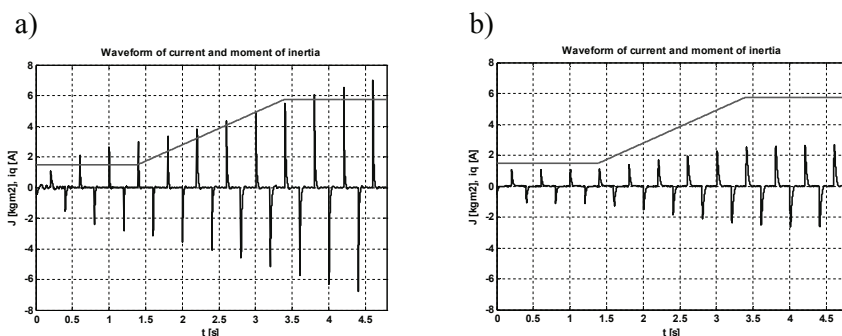
### 4. WYNIKI BADAŃ SYMULACYNYCH

Ważnym procesem jest pierwszy cykl uczenia sztucznej sieci neuronowej, startujący z przypadkowo dobranymi wagami (mogą to być również wagi zerowe). Jak wykazały wcześniejsze badania [6, 7], proces odpowiedzi na skok wartości zadanej prędkości przebiega w takiej sytuacji oscylacyjnie z dużym przeregulowaniem. Można temu przeciwdziałać, i tak, na początku dla prędkości zerowej wprowadzić do układu skok momentu wirtualnego [6, 7], lub użyć sygnału

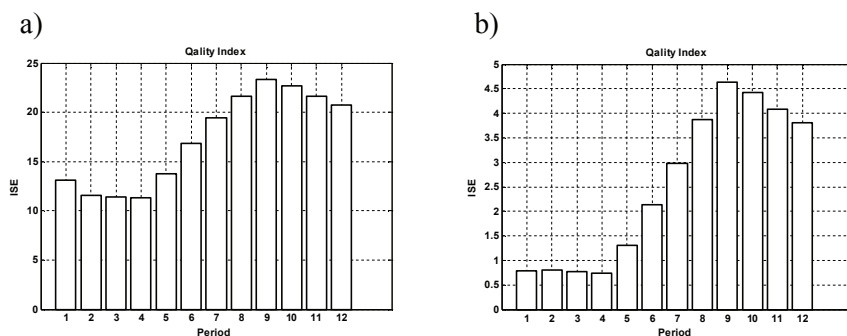
prędkości zadanej z prefiltrem [6, 7]. W poprzednich pracach [6, 7] przedstawiono również wpływ stałej czasowej opóźnienia na uchyb prędkości obrotowej podczas skoku momentu oporowego oraz wpływu strefy histerezy błędu uczenia sieci neuronowej. W niniejszych badaniach przedstawiony zostanie wpływ zmian momentu bezwładności w funkcji czasu dla układu z prefiltrem i bez prefiltrem.



Rys. 3. Przebieg prędkości obrotowej dla układu bez prefiltrowania (a) i z prefiltrem (b) podczas zmiany momentu bezwładności dla sieci wstępnie nauczonej



Rys. 4. Przebieg prądu w osi  $q$  (-) oraz momentu bezwładności  $J$  (-) dla układu bez prefiltrowania (a) i z prefiltrem (b) dla sieci wstępnie nauczonej



Rys. 5. Przebieg wskaźnika jakości ISE układu bez prefiltrowania (a) i z prefiltrem (b) dla sieci wstępnie nauczonej

W przeprowadzonych, i przedstawionych badaniach symulacyjnych przyjęto rozwiązanie, w którym, po wstępnym nauczaniu sztucznej sieci neuronowej współczynniki wagi neuronów zostały zapamiętane w pamięci i przy każdym ponownym uruchamianiu przepisane do regulatora. Algorytm koryguje współczynniki wag w kolejnych krokach, w chwilach występowania błędów regulacji, czyli w trakcie kolejnych procesów przejściowych wywołanych zmianami prędkości zadanej, momentu oporowego oraz zmian momentu bezwładności. Zmiany współczynników wag są jednak zdecydowanie mniejsze, niewywołujące znacznych oscylacji [6, 7]. Na rys. 3 przedstawiono przebieg zmian prędkości obrotowej dla układu bez prefiltrowania (a) i z prefiltrem (b) podczas zmiany momentu bezwładności dla wstępnie nauczonej sieci neuronowej. Proces zmiany momentu bezwładności rozpoczął się w czasie 1,4 s od wartości  $1,5 \text{ kgm}^2$  i trwał do 3,4 s. Zakończył się na wartości  $5,8 \text{ kgm}^2$  (rys. 4). Na rys. 5 przedstawiono wskaźnik jakości ISE (ang. Integral Square Error) uchybu prędkości dla poszczególnych okresów zmian prędkości.

Porównując przebiegi prędkości z rys. 3a i 3b można zauważyć, że układ bez prefiltrowania przejawia tendencję do niewielkich przeregulowań zarówno przed zmianą momentu bezwładności, jak i po zmianie. Wzrost wartości wskaźnika ISE można zaobserwować po rozpoczęciu zmiany momentu bezwładności, a po jej zakończeniu obserwujemy zmniejszenie wartości tego wskaźnika, co świadczy o kontynuowaniu procesu adaptacji. Podobnie wygląda przebieg z prefiltrem, ale przed rozpoczęciem zmian nie zaobserwowano przeregulowania prędkości kątowej, natomiast niewielkie pojawiło się podczas zmiany momentu bezwładności. Po zakończeniu tego procesu, widać wyraźny spadek wskaźnika ISE. Porównując oba wskaźniki (rys. 5a i 5b) można zauważyć również, że wartość tego parametru dla układu z prefiltrem ( $ISE = 0,8$ ) jest zdecydowanie mniejsza w porównaniu z układem bez prefiltrowania ( $ISE = 12$ ), zarówno przed rozpoczęciem procesu zmian momentu bezwładności, podczas jego zmian i po zakończeniu (czas uczenia dla obu układów był taki sam). Natomiast, podczas zmian momentu bezwładności wskaźnik ISE dla układu bez prefiltrowania wzrósł dwukrotnie ( $ISE = 24$ ), natomiast dla układu z prefiltrem można zauważyć ponad pięciokrotny wzrost tej wartości ( $ISE = 4,5$ ). Nadal jednak wskaźnik ten (z prefiltrem) jest ponad pięciokrotnie niższy z układem porównywanym.

## 5. WNIOSKI

Przedstawione wyniki symulacyjne przedstawiają poprawne działanie regulatora neuronowego, uczonego w czasie rzeczywistym. Przedstawiona koncepcja adaptacyjnego regulatora neuronowego pozwala uniezależnić właściwości dynamiczne regulacji prędkości kątowej od zmian parametrów serwonapędu, a w szczególności zmian momentu bezwładności. Przyjęta metoda uczenia sieci neuronowej RPROP umożliwia uczenie sztucznej sieci neuronowej

„on-line” czyli w czasie rzeczywistym podczas zachodzących szybkich procesów przejściowych. Zaproponowany regulator neuronowy posiada interesujące właściwości, które zostaną przedstawione w dalszych pracach.

### LITERATURA

- [1] Ellis G., Control System Design Guide. Third Edition: Using Your Computer to Understand and Diagnose Feedback Controllers, *Elsevier*, Academic Press, 2004.
- [2] Grzesiak L., Meganek V., Sobolewski J., Ufnalski B., DTC\_SVM Drive with ANN-based Speed Controller, PELINCEC Conference, Warsaw University of Technology, 2005, CD.
- [3] Grzesiak L. M., Meganek V., Sobolewski J., Ufnalski B., On-line Trained Neural Speed Controller with Variable Weight Update Period for Direct-Torque-Controller AC Drive, EPE-PEMC Conference, Portoroz, 2006, CD.
- [4] Orłowska-Kowalska T., Szabat K., Control of the Drive System with Stiff and Elastic Coupling using Adaptive Neuro-Fuzzy Approach, IEEE Trans. On Industrial Electronics, vol. 51, No. 4, 2007, pp. 228-240.
- [5] Ossowski S., Sieci neuronowe do przetwarzania informacji, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 2000.
- [6] Pajchrowski T., Zawirski K., Adaptive Neural Speed Controller for PMSM Servodrive with Variable Parameters, Proceedings of EPE-PEMC 2012 ECCE Europe Conference and Exposition Conference, 4th to 6th September 2012, Novi Sad, Serbia.
- [7] Pajchrowski T., Zawirski K., Application of artificial neural network for adaptive speed control of PMSM drive with variable parameters, Proceedings of XXII Symposium on Electromagnetic Phenomena in Nonlinear Circuits, EPNC'2012, Pula, Croatia, 26 -29 June 2012.
- [8] Pajchrowski T., Zawirski K., Application of Artificial Neural Network to Robust Speed Control of Servodrive. IEEE Transaction on Industrial Electronics, Vol.54, No.1, February 2007, pp.200-207.
- [9] Pajchrowski T., Zawirski K., Robust speed and position control based on neural and fuzzy techniques, EPE 2007, Aalborg, Power Electronics and Applications, 2007 European Conference on 2-5 Sept. 2007, E-ISBN 978-92-75815-10-8.
- [10] Riedmiller M., Braun H., A direct adaptive method for faster backpropagation learning: The RPROP algorithm, IEEE International Conference on Neural Networks pp. 586-591, vol. 1, 28 March – 1 April, 1993.
- [11] Rutkowski L., Metody i techniki sztucznej inteligencji, Wydawnictwo Naukowe PWN, Warszawa 2005.

### ADAPTIVE PERMANENT MAGNET SYNCHRONOUS MOTOR CONTROL

This paper presents the results of simulation of adaptive speed controller using an artificial neural network for permanent magnet synchronous motor drive. Discusses the structure of the artificial neural network and the method of learning the speed controller in real time. Model system has been developed in Matlab. The controller parameters are optimized on-line by RPROP algorithm. The simulation results illustrate the proper operation of the adaptive speed control to change the parameters of the drive system, such as the moment of inertia.